UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/674,974	09/30/2003	Yen-Fu Chen	AUS920030588US1	4970

35525 7590 11/28/2008

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
TIMBLIN, ROBERT M	
ART UNIT	PAPER NUMBER
2167	

NOTIFICATION DATE	DELIVERY MODE
11/28/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptonotifs@yeeiplaw.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/674,974
Filing Date: September 30, 2003
Appellant(s): CHEN ET AL.

Rudolf O. Siegesmund
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 8/27/2008 appealing from the Office action mailed 3/27/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,609,133	Ng et al.	8-2003
7,016,906	Janzig et al.	3-2006
2002/0120685	Srivastava et al.	8-2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 3-11, 13-20, 22-29, and 32-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ng et al. (Ng hereinafter) (US 6,609,133 B2) in view of Janzig et al. (Janzig hereafter U.S. Patent 7,016,906) and further in view of Srivastava et al. (Srivastava hereinafter) (US 2002/0120685 A1).

With respect to claim 1, Ng teaches a method for validating data in a backend driven environment, the method comprising:

creating an XML Schema for a database (col. 4 line 1-7 and line 35-42, col. 5 line 45-50, and figure 3), wherein the XML Schema contains a plurality of rules for validating (col. 5 line 20-33) a plurality of data in the database (col. 9 line 8-28, figure 12, col. 4 line 36-42, and col. 5 line 45-50);

copying the database to a hash table (col. 5 line 60-67, col. 6 line 19-27, and col. 11 line 10-58);

determining if the database and the hashtable are not identical (col. 7 line 28-41, col. 8 line 10-27, and figure 10); and

when the database and the hash table are not identical, creating a new XML Schema (col. 8 line 29-50);

wherein the step of creating a new XML Schema includes automatically updating the plurality of rules (col. 4 line 29-51); and

Ng fails to expressly teach wherein a new XML schema is created only when a determination is made that the database and the hashtable are not identical.

Janzig, however, teaches a new XML schema is created only when a determination is made that the database and the hashtable are not identical (col. 4 line 42-54, col. 9 lines 14 and 30-35, and figure 19) to determine that the structure of the database has changed and thereby creating a new schema. Furthermore, Janzig would have given Ng explicit validation rules (i.e. Janzig at col. 10, describes two tables with valid data types accepted in a database) for validating data and updating those rules when the new schema is created.

In the same field of endeavor, (i.e. creating database schemas and preserving changes thereto), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the teachings of Janzig would have given Ng a new schema when the database is changed (as taught by Janzig at col. 2 line 30-30). Ng is concerned with creating new schemas as databases evolve (i.e. Ng at col. 2 lines 38-43) and comparing a database to a hashtable (i.e. col. 7 line 28-41) to determine if they are

identical (in other words to determine change in a database). Therefore it would have been obvious to combine Janzig's teachings of creating a new schema when Ng determines database changes to keep the database and schema consistent.

Ng also fails to expressly teach designating a query interval and upon the occurrence of a query interval, comparing the database to the hash table, and the use of an XML Schema for validating data.

Srivastava however, teaches designating a query interval and upon the occurrence of a query interval, comparing the database to the hash table (0068-0069) for specifying when to perform an update check. Furthermore, Srivastava teaches the use of an XML schema (paragraph 0010) for validating data.

In the same field of endeavor, (i.e. data processing), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Srivastava would have provided Ng's system with ensuring system integrity with use of an XML schema. Further, Srivastava's teaching would have Given Ng a query interval for checking updates for the benefit of efficiently determining when a data source has been modified (as needed in Ng at col. 3 line 35-39 and 53-56).

With respect to claim 3, Ng fails to teach resetting the query interval.

Srivastava, however teaches resetting the query interval (0068). The motivation for combination can be equally applied from claim 1. The rejection of claim 3 equally applies well to claim 22 and 32.

With respect to claim 4, Ng teaches the method of claim 1 further comprising: deleting the hashtable and saving the database as a new hashtable (col. 8 line 42-58).

With respect to claim 5, Ng teaches the method of claim 1 further comprising: storing the new XML Schema in a web server's virtual root (col. 4 line 53-col. 5 line 5 and figure 1).

With respect to claim 6, Ng teaches 6. The method of claim 1 wherein a limited number of tables from the database are copied to the hashtable (col. 6 line 1-28); and

wherein upon the occurrence of a query interval, the database tables are compared to the tables in the hashtable (col. 8 line 10-28).

With respect to claim 7, Ng teaches the method of claim 1 wherein a database metadata is copied to the hashtable (col. 5 line 60-67); and

wherein upon the occurrence of a query interval, the database metadata is compared to the metadata in the hashtable (col. 8 line 10-28).

With respect to claim 8, Ng fails to teach notifying a registered party of an update to the XML schema.

Srivastava, however, teaches notifying a registered party of an update to the XML Schema (0068 and 0447).

In the same field of endeavor, (i.e. data processing), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the notification method of Srivastava would have given Ng a way to notifying users of change for further providing an indication of modification which is needed by Ng in col. 3 line 53-55. This rejection equally applies well to claim 18, 27, and 37.

With respect to claim 9, Ng fails to teach using a database trigger to indicate a change in the database.

Srivastava, however, teaches using a database trigger to indicate a change in the database (0448) for executing services upon the occurrence of events.

In the same field of endeavor, (i.e. data processing), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the teaching of Srivastava would have given Ng a further efficient way to detect modifications. This rejection equally applies well to claim 19, 28, and 38).

With respect to claim 10, Ng teaches a first method for validating proposed additions to a database comprising:

accessing an XML Schema stored in a web server's virtual root (col. 4 line 53-col. 5 line 5 and figure 1);

submitting the data to a database (col. 5 line 28-31); validating the data (col. 5 line 28-31), and adding the validated data to the database (e.g. in col. 5, line 28-31 Ng teaches determining if a field can accept a null value, thereby suggesting validation of incoming data and upon acceptance, adding it to the database),

wherein the XML Schema is created by a second method comprising:

creating an XML schema for a database (col. 4 line 1-7 and line 35-42, col. 5 line 45-50, and figure 3);

copying the database to a hash table (col. 5 line 60-67, col. 6 line 19-27, and col. 11 line 10-58).

Ng fails to expressly teach when the database and the hashtable are not identical, creating a new XML Schema.

Janzig, however, teaches when the database and the hashtable are not identical, creating a new XML Schema (col. 4 line 42-54, col. 9 lines 14 and 30-35, and figure 19) to determine that the structure of the database has changed and thereby creating a new schema. Furthermore, Janzig would have given Ng explicit validation rules (i.e. Janzig at col. 10, describes two tables with valid data types accepted in a database) for validating data and updating those rules when the new schema is created.

In the same field of endeavor, (i.e. creating database schemas and preserving changes thereto), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the teachings of Janzig would have given Ng a new schema when the database is changed (as taught by Janzig at col. 2 line 30-30). Ng is concerned

with creating new schemas as databases evolve (i.e. Ng at col. 2 lines 38-43) and comparing a database to a hashtable (i.e. col. 7 line 28-41) to determine if they are identical (in other words to determine change in a database). Therefore it would have been obvious to combine Janzig's teachings of creating a new schema when Ng determines database changes to keep the database and schema consistent.

Ng also fails to expressly teach designating a query interval and upon the occurrence of a query interval, comparing the database to the hash table, and the use of an XML Schema for validating data.

Srivastava however, teaches designating a query interval and upon the occurrence of a query interval, comparing the database to the hash table (0068-0069) for specifying when to perform an update check. Furthermore, Srivastava teaches the use of an XML schema (paragraph 0010) for validating data.

In the same field of endeavor, (i.e. data processing), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Srivastava would have provided Ng's system with ensuring system integrity with use of an XML schema. Further, Srivastava's teaching would have Given Ng a query interval for checking updates for the benefit of efficiently determining when a data source has been modified (as needed in Ng at col. 3 line 35-39 and 53-56).

With respect to claim 11 Ng teaches the first method of claim 10 further comprising: creating an XML Schema for a database (col. 4 line 1-7 and line 35-42, col. 5 line 45-50, and figure 3).

With respect to claim 13, Srivastava teaches the first method of claim 10 wherein the second method further comprises: when the database and the hashtable are identical, resetting the query interval [0068] and repeating the steps in claim 10. The motivation for combination can equally apply from the rejection of claim 10.

With respect to claim 14, Ng teaches the method of claim 10 wherein the second method further comprises: deleting the hashtable and saving the database as a new hashtable (col. 8 line 42-58).

With respect to claim 15, Ng teaches the method of claim 10 wherein the second method further comprises: storing the new XML Schema in a web server's virtual root (col. 4 line 53-col. 5 line 5 and figure 1).

With respect to claim 16, Ng teaches the first method of claim 10 wherein the second method further comprises: wherein a limited number of tables from the database are copied to the hashtable; and wherein upon the occurrence of a query interval, the database tables are compared to the tables in the hashtable (col. 6 line 1-28).

With respect to claim 17, Ng teaches the first method of claim 10 wherein the second method further comprises: wherein a database metadata is copied to the hashtable (col. 5 line 60-67); and

wherein upon the occurrence of a query interval, the database metadata is compared to the metadata in the hashtable (col. 8 line 10-28).

With respect to claim 20, Ng teaches a program product operable on a computer, the program product comprising:

a computer-usable medium (figure 1);

wherein the computer usable medium comprises instructions contained in the program product comprising:

instructions for creating an XML Schema for a database (col. 4 line 1-7 and line 35-42, col. 5 line 45-50, and figure 3);

wherein the XML Schema contains a plurality of rules for validating (col. 5 line 20-33) a plurality of data in the database (col. 9 line 8-28, figure 12, col. 4 line 36-42, and col. 5 line 45-50);

instructions for copying the database to a hashtable (col. 5 line 60-67, col. 6 line 19-27, and col. 11 line 10-58);

instructions for determining if the database and the hashtable are identical (col. 7 line 28-41, col. 8 line 10-27, and figure 10); and

responsive to a determination that that database and the hashtable are identical (col. 7 line 28-41, col. 8 line 10-27, and figure 10), instructions for performing additional steps comprising: instructions for creating a new XML Schema (col. 8 line 29-50);

wherein the instructions for creating a new XML Schema cause the computer to automatically update the plurality of rules (col. 4 line 29-51).

Ng fails to expressly teach wherein a new XML schema is created only when a determination is made that the database and the hashtable are not identical.

Janzig, however, teaches a new XML schema is created only when a determination is made that the database and the hashtable are not identical (col. 4 line 42-54, col. 9 lines 14 and 30-35, and figure 19) to determine that the structure of the database has changed and thereby creating a new schema. Furthermore, Janzig would have given Ng explicit validation rules (i.e. Janzig at col. 10, describes two tables with valid data types accepted in a database) for validating data and updating those rules when the new schema is created.

In the same field of endeavor, (i.e. creating database schemas and preserving changes thereto), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the teachings of Janzig would have given Ng a new schema when the database is changed (as taught by Janzig at col. 2 line 30-30). Ng is concerned with creating new schemas as databases evolve (i.e. Ng at col. 2 lines 38-43) and comparing a database to a hashtable (i.e. col. 7 line 28-41) to determine if they are identical (in other words to determine change in a database). Therefore it would have been obvious to combine Janzig's teachings of creating a new schema when Ng determines database changes to keep the database and schema consistent.

Ng also fails to expressly teach designating a query interval and upon the occurrence of a query interval, comparing the database to the hash table, and the use of an XML Schema for validating data.

Srivastava however, teaches designating a query interval and upon the occurrence of a query interval, comparing the database to the hash table (0068-0069) for specifying when to perform an update check. Furthermore, Srivastava teaches the use of an XML schema (paragraph 0010) for validating data.

In the same field of endeavor, (i.e. data processing), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Srivastava would have provided Ng's system with ensuring system integrity with use of an XML schema. Further, Srivastava's teaching would have Given Ng a query interval for checking updates for the benefit of efficiently determining when a data source has been modified (as needed in Ng at col. 3 line 35-39 and 53-56).

With respect to claim 23, Ng teaches the program product of claim 20 further comprising: when the database and the hashtable are not identical, instructions for deleting the hashtable and saving the database as a new hashtable (col. 8 line 42-58).

With respect to claim 24, Ng teaches the program product of claim 20 further comprising: when the database and the hashtable are not identical, instructions for storing the new XML Schema in a web server's virtual root (col. 4 line 53-col. 5 line 5 and figure 1).

With respect to claim 25, Ng teaches the program product of claim 20 wherein a limited number of tables from the database are copied to the hashtable (col. 6 line 1-28); and

wherein upon the occurrence of a query interval, the database tables are compared to the tables in the hashtable (col. 8 line 10-28).

With respect to claim 26, Ng teaches the program product of claim 20 wherein a database metadata is copied to the hashtable (col. 5 line 60-67); and

wherein upon the occurrence of a query interval, the database metadata is compared to the metadata in the hashtable (col. 8 line 10-28).

With respect to claim 29, Ng teaches a first program product operable on a computer, the program product comprising:

a computer-usable medium (figure 1);

wherein the computer usable medium comprises instructions contained in the program product comprising:

instructions for accessing an XML Schema stored in a web server's virtual root (col. 4 line 53-col. 5 line 5 and figure 1);

wherein the XML Schema contains a plurality of rules for validating (col. 5 line 20-33) a plurality of data in the database (col. 9 line 8-28, figure 12, col. 4 line 36-42, and col. 5 line 45-50);

instructions for checking the validity of data using the XML Schema (col. 5 line 28-31), submitting the data to a database (col. 5 line 28-31); validating the data (col. 5 line 28-31), and adding the validated data to the database (e.g. in col. 5, line 28-31 Ng teaches determining if a field can accept a null value, thereby suggesting validation of incoming data and upon acceptance, adding it to the database)

wherein the XML Schema is created by a second program product comprising:

instructions for determining if the database and the hashtable are identical (col. 7 line 28-41, col. 8 line 10-27, and figure 10);

when the database and the hashtable are not identical (col. 8 line 29-50), instructions for creating a new XML Schema (col. 8 line 29-50):

wherein the step of creating a new XML Schema includes automatically updating the plurality of rules (col. 4 line 29-51); and

Ng fails to expressly teach when the database and the hashtable are not identical, creating a new XML Schema.

Janzig, however, teaches when the database and the hashtable are not identical, creating a new XML Schema (col. 4 line 42-54, col. 9 lines 14 and 30-35, and figure 19) to determine that the structure of the database has changed and thereby creating a new schema. Furthermore, Janzig would have given Ng explicit validation rules (i.e. Janzig at col. 10, describes two tables with valid data types accepted in a database) for validating data and updating those rules when the new schema is created.

In the same field of endeavor, (i.e. creating database schemas and preserving changes thereto), it would have been obvious to one of ordinary skill in the data

processing art at the time of the present invention to combine the teachings of the cited references because the teachings of Janzig would have given Ng a new schema when the database is changed (as taught by Janzig at col. 2 line 30-30). Ng is concerned with creating new schemas as databases evolve (i.e. Ng at col. 2 lines 38-43) and comparing a database to a hashtable (i.e. col. 7 line 28-41) to determine if they are identical (in other words to determine change in a database). Therefore it would have been obvious to combine Janzig's teachings of creating a new schema when Ng determines database changes to keep the database and schema consistent.

Ng also fails to expressly teach designating a query interval and upon the occurrence of a query interval, comparing the database to the hash table, and the use of an XML Schema for validating data.

Srivastava however, teaches designating a query interval and upon the occurrence of a query interval, comparing the database to the hash table (0068-0069) for specifying when to perform an update check. Furthermore, Srivastava teaches the use of an XML schema (paragraph 0010) for validating data.

In the same field of endeavor, (i.e. data processing), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Srivastava would have provided Ng's system with ensuring system integrity with use of an XML schema. Further, Srivastava's teaching would have Given Ng a query interval for checking updates for the benefit of efficiently determining when a data source has been modified (as needed in Ng at col. 3 line 35-39 and 53-56).

With respect to claim 33, Ng teaches the first program product of claim 29 wherein the second program product further comprises: instructions for deleting the hashtable and saving the database as a new hashtable (col. 8 line 42-58).

With respect to claim 34, Ng teaches the first program product of claim 29 wherein the second program product further comprises: instructions for storing the new XML Schema in a web server's virtual root (col. 4 line 53-col. 5 line 5 and figure 1).

With respect to claim 35, Ng teaches the first program product of claim 30 wherein a limited number of tables from the database are copied to the hashtable (col. 6 line 1-28); and

wherein upon the occurrence of a query interval, the database tables are compared to the tables in the hashtable (col. 8 line 10-28).

With respect to claim 36, Ng teaches the first program product of claim 30 wherein a database metadata is copied to the hashtable (col. 5 line 60-67); and

wherein upon the occurrence of a query interval, the database metadata is compared to the metadata in the hashtable (col. 8 line 10-28).

(10) Response to Argument

Appellant's arguments filed in the Appeal Brief dated 8/27/2008 (herein 'Brief') have been fully considered but they are not persuasive.

(1) On page 10 of the Brief and in regards to claims 1, 10, 20, and 29, Appellant argues that these claims distinguish over the prior art. Examiner disagrees given the following:

In particular, at the top of page 10, Appellant argues that Claim 1 is distinguished because Ng does not disclose an XML Schema. The Examiner's response:

Substantially, the Appellant argues that the term "XML Schema" in the present disclosure and the term "schema" used by Ng is different. Appellant further states that they define an XML Schema as a computer file containing rules for validating data "to determine if the data meets the requirements of the database records, tables and fields." The Examiner submits that Ng's teachings corresponds to Appellant's definition of XML Schema and therefore teach at least this limitation.

Specifically, Ng discloses that "Object-relational mapping tool 114 reads database 118 to examine its schema, constructs database data structure 115 to reflect this schema, generates an object model 116 based on database data structure 115, and then creates source code 120 based on object model 116 (Ng, col. 5 line 20-24). Ng then describes that the source code may essentially comprise methods to control the

attributes of a field by stating "the programmer may... change the attributes of a field (e.g. the type or whether it can accept a null value)." With these excerpts, it can be seen that the source code based on the object model (which is based on a database data structure that reflects the schema) contains rules for validating data. In other words, Ng's source code (that essentially reflects the schema; e.g. see also Ng, col. 4 line 35) contains rules that specify what values (e.g. the type or null) *can* be accepted for a particular field. In this interpretation, these values are validated for the fields by the source code because they are checked to determine if they meet certain requirements (i.e. if it is the correct type or if a null value can be accepted) for acceptance into the fields. Because Ng's schema is seen to contain source code that specifies what values can be accepted in a field, Ng's schema meets Appellant's definition of an XML Schema which is a computer file containing rules for validating data "to determine if the data meets the requirements of the database records, tables and fields."

Furthermore, Appellant submits in the Brief that "[B]y schema, Ng means the logical structure of the database" (page 10, argument (1) of the brief). The Examiner responds that even if Ng's schema is the logical structure of a database that data in the database would be "validated" against this logical structure. Put another way, data would have to conform (i.e. the data would have to be validated/checked against) to the schema in order to maintain proper structure and consistent interrelationships (col. 2 line 12).

Also, in respect to this argument, Appellant states that "[M]oreover, Ng does not automatically update a schema – rather Ng takes an updated schema and then creates a new source code for the database." The Examiner first notes that this limitation is not found explicitly in the claim; rather, what is claimed is "creating a new XML Schema; wherein the creating a new XML schema includes *automatically updating the plurality of rules*..." Nonetheless, the Examiner submits that Ng teaches updating an object model in drawing reference 506, fig 5 (where the object model is based on the database data structure that reflects the schema, Ng, col. 5 line 20-23). Because the source code in Ng contains the rules for validating data as detailed above, and that new source code is created in response to a new schema (Ng, col. 4 line 39-45), that the rules for validation are automatically updated.

(2) Appellant argues on page 10 of the Brief that Claim 1 is distinguished because Ng does not disclose "validating."

Examiner disagrees for the same rationale as given in the preceding paragraphs. In summary, Ng teaches source code that shares a correspondence with the database schema (i.e. the source code is created based on the object model that is based on a database data structure that reflects the database schema; col. 5 lien 20-25, Ng). That is, Ng teaches the source code specifies at least the type of values that a field can accept (col. 5 line 30-31).

Appellant defines "validate" as checking "data against an XML Schema to determine if the data meets the requirements of the database records, tables and

fields." As the definition is not explicitly found in the claims, Ng teaches the claim as defined in the disclosure. As in the example given above (e.g. col. 5 line 30-31; i.e. specifying what values can be accepted), Ng's source code which is substantially derived from the schema (e.g. see col. 2 line 9-11 wherein a database schema is read and source code is generated and col. 2 line 36-37 wherein source code is generated with classes that reflect the structure of the database at that time), contains "rules" that data is checked against and therefore teaches "validating". In the above example, the source code of the schema specifies and therefore *requires* the type of data that can be admitted into a field and therefore teaches a requirement of the database fields.

In the first paragraph of page 11 in the Brief, Appellant asserts that the Examiner relies on inherency of Ng's disclosure to teach validation. The Examiner disagrees and maintains that Ng expressly teaches validation by the disclosure of a schema. Specifically, the Examiner has supported with evidence that Ng teaches validation at least by source code, derived effectively from the schema to specify and require what type of data can be included in a database field (e.g. col. 5 line 30-31). Furthermore the Examiner submits that Ng suggests validation by describing a schema that is read to generate source code that contains methods that act upon data to get and set values (col. 2 line 9-21). Again, this source code derived from the schema specifies rules and requirements for the database (e.g. col. 5 line 30-31) and therefore teaches validation of data in the database.

Moreover, the Appellant appears to argue (last paragraph of page 11 in the Brief) that Srivastava does not teach "validation". To reiterate from the above rejection, Srivastava was relied upon to teach an "XML" schema wherein Ng does not appear to teach 'XML'. Srivastava further teaches that an XML Schema is used for validation (e.g. paragraph 0010). That is, Srivastava teaches validating a service description against a Service Descriptor Schema which may take the form of an XML schema such as a DTD *before* it is stored in the registry (to meet Appellant's argument on page 10 of the Brief). In other words, Srivastava explicitly uses an XML schema to validate data in paragraph 0010.

(3) In the beginning of page 12 of the Brief, Appellant argues that the cited art does not teach **a)** when the database and the hashtable are not identical, creating a new XML Schema and **b)** wherein the new XML Schema is created only when a determination is made that the database and the hashtable are not identical. The Examiner disagrees given the following:

For **a)**, the Examiner disagrees because Ng is seen to teach this aspect. Specifically, Ng teaches the use of a hashtable to determine database changes. For example, Ng teaches the use of a hashtable in comparison to a database (col. 8 line 17-20) to determine changes. Put another way, a hash table of the two database data structures is compared (see also figure 10 and col. 8 line 16-20, Ng) to determine if any changes (i.e. if they are identical) are made.

Subsequently, from this determination, Ng teaches the creation of a new XML Schema (e.g. col. 8 lines 29-50). That is, Ng teaches that the changes are isolated and the object model is updated (see also figure 5). In col. 8 line 34-36, Ng teaches that new source code, reflecting both the schema changes and customizations can be generated from the object model (Ng, col. 8 line 34-37). The Examiner submits that at least this teaching maps to the current claims which define the creation of a new XML Schema includes automatically updating the plurality of rules. In other words, when changes are determined using the hashtable (col. 8 line 10-27), the source code (which has been described to contain "rules") is updated accordingly (e.g. drawing reference 508, fig. 5 - Ng).

For **b)**, the Examiner disagrees because at least Janzig teaches this aspect. That is, Janzig teaches the finding of a database change (i.e. when the database and the hashtable are not identical), and only then, creating a new schema. See figure 19 which illustrates the finding of change (211) and determining that a change has taken place (212) a new schema is written (221). Otherwise, if there has been no change, then Janzig teaches to do nothing.

The Examiner submits that it would have been obvious to combine this teaching of Janzig to Ng for creating a new schema only when a database change is detected (i.e. when Ng compares a hashtable to a database) for having a new schema only when needed (i.e. as databases evolve; col. 2 lines 38-43, Ng) thus saving development time (as needed by Ng, col. 4 line 15-16).

Further on page 13 of the Brief (first indented paragraph), Appellant argues that the cited art does not teach "upon the occurrence of a query interval, comparing the database to the hashtable" because Srivastava fails to teach "comparing the database to the hashtable." In response, the Examiner submits that Srivastava teaches a periodic search for updated service information (i.e. data) and a specified frequency with which the update check is to be performed (0068, Srivastava) to teach the claimed query interval. The Examiner also submits that Ng teaches a comparison of a database to a hashtable by teaching the claimed limitation "determining if the database and the hashtable are not identical" (col. 7 line 28-41, col. 8 line 10-27, and figure 10 as seen in the above rejection). For example, Ng teaches that all the fields in the hash table of the two database data structures are used to determine any changes (col. 8 line 17-20). In at least this citation, Ng teaches a comparison of the database to the hash table by ascertaining if they are identical.

Therefore, the Examiner submits that Srivastava's query interval (used to determine changes) combined with Ng's determination if a database and hashtable are identical (i.e. a comparison using a hashtable to determine changes) would have allowed Ng to specify when to compare a database to a hashtable in order to determine changes for the benefit of efficiently determining when a data source has been modified. Accordingly, the combination of Ng and Srivastava teach "upon the occurrence of a query interval, comparing the database to the hashtable."

Argument d. (page 14 of the Brief) in regards to Claims 3, 13, 22, and 32:

Appellant argues that Srivastava does not teach resetting the query interval. The Examiner disagrees and maintains that Srivastava teaches the specification (setting) of a query interval (0068) by teaching a periodical search for updated information. The Examiner submits that at least once the period is satisfied, and the update check is performed, that the query interval would be essentially reset so that it may occur again. Put another way, the update check is periodically reset so that the frequency of update checks can be performed.

Argument e. (page 14 of the Brief) in regards to Claims 4, 14, 23, and 33:

Appellant argues that Ng is silent as to deleting a hashtable and saving the database as a new hashtable. The Examiner disagrees because Ng teaches the deletion of entries in a hash table (col. 8 line 57-58) and therefore describes that the hash table can be deleted when no longer needed. In other words, if a column of a database is found to be deleted, then the corresponding hash table is deleted as well. Ng further teaches saving the database as a new hash table (e.g. col. 5 line 61-67 and col. 6 line 18-28) wherein it is disclosed that a hashtable contains values for a database.

Examiner further submits that as the definition of a hash table is not found in the claims, this aspect has been given a reasonably broad interpretation in accordance with MPEP 2111.

Argument f. (page 14 of the Brief) in regards to Claims 5, 15, 24, and 34:

Appellant argues that Ng is silent to teaching storing the new XML Schema in a web server's virtual root." Examiner disagrees because the schema in Ng can be stored (col. 4 line 53-col. 5 line 5) in a computing device (100, figure 1) attached to a network (fig 1, reference 102) to suggest a server. As the schema is stored in a computing device that is capable of providing through a network, that computing device is seen as a server, and further more a "virtual root" (or source) to store the schema.

Furthermore, the Examiner submits that Appellant is features not found in the claim by supplementing limitations from the specification into the claims (i.e. "Indeed, the advantage of storing the XML Schema in the web server's virtual root is explained in the specification" – Appellant, page 14 of the Brief, **argument f.**). In response, the Examiner submits that as the claim is broadly stated, the limitations therein have been given their broadest reasonable interpretation in accordance with MPEP 2111.

Argument g. (page 14-15 of the Brief) in regards to Claims 6, 16, 25, and 35:

Appellant argues here that Ng's hash table is not the same as Appellant's defined term hashtable. Examiner disagrees and maintains that as the claim (e.g. claim 1) merely suggests the hashtable to store a copy of the database, that Ng discloses this aspect. Specifically, Ng teaches copying a database to a hashtable (e.g. col. 6 line 19-27) wherein their hash table contains an entry for each field in a customer table. In this sense, an "entry for each field in a customer table" is seen as a copy of that table to a hashtable.

Argument h. (page 15 of the Brief) in regards to Claims 7, 17, 26, 36:

Appellant argues that Ng fails to disclose the limitations as set forth above. The Examiner submits that the rebuttal set forth above applies equally well in response to this argument.

Argument i. (page 15 of the Brief) in regards to Claims 8, 18, 27 and 37:

Appellant argues the definition of XML Schema distinguishes over the prior art. The Examiner disagrees at least for the rationale given in response to arguments (1)-(2)

Argument j. (page 15 of the Brief) in regards to Claim 9:

Appellant argues that Srivastava is silent to "a database trigger" and using such a trigger to "indicate a change in the database." The Examiner disagrees and maintains that Srivastava teaches this element by defining triggers in [0448]. Srivastava further teaches triggers for indicating changes in a database by describing a query interval (0068) that detects and checks for updates. With this, the services engine is a trigger service that indicates changes in the database.

Argument k. (page 15 of the Brief) in regards to Claim 11:

Appellant argues that this limitation is distinguished for the reasons stated above. The Examiner submits the rationale set forth above applies equally well to this claim.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/ROBERT TIMBLIN/

Examiner, Art Unit 2167

Conferees:

John Cottingham

/John R. Cottingham/

Supervisory Patent Examiner, Art Unit 2167

/Eddie C. Lee/

Supervisory Patent Examiner, TC 2100

Eddie Lee